Table des matières

Table des matières.. 1
I. Introduction.. 2
II. Présentation du projet.. 3
A. Bilan d'énergie et équation différentielle.. 3
B. Equation en régime stationnaire et modèle sans pertes...4
C. Régime permanent dépendant du temps: onde sinusoïdale..4
III. Explication sur le logiciel LabVIEW.. 6
A. Historique de LabVIEW.. 6
B - Un logiciel dédié à la programmation instrumentale..6
C. Avantages de LabVIEW.. 7
D. Comment marche LaVIEW?..8
IV. Algorithme complet du programme... 8
A. Algorithme général.. 8
B. Sous algorithmes.. 9
V. Description des parties importantes du programme.. 12
A. Principes de base...12
B. Détails des différentes tâches... 16
VI. Manuel d'utilisation...21
A. Introduction et vue d'ensemble 21
B. Démarrage...22
C. Interface principale...24
D. Interface secondaire.. 24
E. Menu Configuration.. 24
F. Utiliser le logiciel.. 25
G. Assistance technique...25
Conclusion...25

I. Introduction

La troisième année de cycle ingénieur à Polytech'Nice-Sophia, département électronique, contient dans son enseignement, pour l'option CCS (Conception Circuits et Systèmes), un cours d'instrumentation. Ce cours se compose de deux parties: la première partie consiste en l'apprentissage du langage de programmation graphique LabVIEW et la seconde est la réalisation d'un projet en binôme.

Le sujet que nous avons choisi est la «propagation d'une onde thermique dans une barre». Ce projet consiste à créer un outil informatique qui permet de calculer la conductivité thermique de différents matériaux: acier, cuivre et aluminium. Cet outil doit simplifier la tâche d'un étudiant lors d'un TP de thermodynamique. Ainsi, ce dernier, en chauffant simplement la barre par une résistance chauffante, peut facilement obtenir, à l'aide de l'interface graphique, les courbes d'évolution de la température en fonction du temps et de la distance des capteurs. La conductivité lui est aussi donnée; donc l'étudiant pourra se concentrer sur l'analyse des résultats grâce au gain de temps fourni par le logiciel créé sous LabVIEW.

Le rapport sera constitué de quatre parties:

- ✓ la première sera dédiée à la description du projet en développant la partie théorique.
- ✓ la seconde donnera des explications sur le logiciel utilisé pour la programmation: LabVIEW.
- ✓ La troisième dévoilera l'algorithme complet du programme.
- ✓ la suivante décrira les points importants du programme.
- ✓ la dernière présentera un manuel d'utilisation du programme et des accessoires.

II. Présentation du projet

L'étudiant dispose d'une barre en acier inoxydable sur laquelle des thermocouples sont placés tous les 10cm. Deux autres barres en aluminium et en cuivre sont aussi disponibles afin de vérifier quel matériau est le plus conducteur de chaleur. Ces barres sont soumises à un chauffage qui peut aussi bien être continu ou dépendant du temps. Le chauffage est effectué par une résistance chauffante de 50W.

Une modélisation du problème est nécessaire avant de commencer à programmer. Ce modèle, qui est donné par la théorie, peut être simplifié selon que l'on se place en régime permanent ou stationnaire.

A. Bilan d'énergie et équation différentielle

Soit une barre placée selon un axe Ox. La température de cette barre dépend de deux paramètres: de la distance et du temps. En faisant un bilan énergétique sur un morceau dx de la barre, on obtient l'équation différentielle donnant l'évolution de T(x,t):

$$\delta T^2/\delta x^2 - (\rho Cp/\lambda)\,\delta T/\delta t - 2\mathrm{h}(T - Ta)/\lambda r = 0$$

Où λ est le coefficient de conductivité thermique donné par la loi de Fourier à une dimension:

$$j = -\lambda\,\delta T/\delta x$$

Enfin *h(T-Ta)* modélise les pertes radiales par une densité de flux thermique (en W/m^2), *h* étant le coefficient d'échange thermique entre la barre et le milieu extérieur à la température ambiante constante *Ta*.

B. Equation en régime stationnaire et modèle sans pertes

Afin de simplifier l'équation différentielle, on émet deux hypothèses. On suppose qu'on est en régime stationnaire, c'est-à-dire que la température ne varie pas en fonction du temps (soit $\delta T/\delta t=0$). De plus, on se place dans le cas où le modèle n'a pas de pertes d'énergie avec l'extérieur donc $h=0$. L'équation à résoudre devient:

$$\delta^2 T/\delta x^2=0$$

Cette équation du second degré a pour solution:

$$T(x)=-jx/\lambda+T(0)$$

On remarque qu'il s'agit de l'équation d'une droite. Ainsi, en calculant la pente de cette droite ($-j/\lambda$), on obtient directement la valeur du coefficient de conductivité thermique λ puisque le paramètre j est constant. En effet, j est calculé à partir de la puissance de la résistance chauffante à travers la surface de la barre soit:

$$j=P/(2\Pi*R*(R+H))$$

On obtient ainsi pour les trois barres les valeurs suivantes du paramètre j:

Acier inoxydable	Cuivre	Aluminium
j = 776,37 W/m²	j = 628,76 W/m²	j= 665,86 W/m²

C. Régime permanent dépendant du temps: onde sinusoïdale

En régime permanent, on fait l'approximation suivante: l'onde thermique issue de la résistance chauffante qui se propage dans la barre est de forme sinusoïdale de pulsation ω. Une solution particulière de l'équation différentielle du bilan d'énergie sera:

$$T(x,t)=A\exp[i(\omega t-kx)]+B(x)$$

B(x) étant donné en régime stationnaire, il est alors possible de déterminer

l'expression analytique de la conductivité thermique en posant $k = \mu - i\nu$. L'expression de la température devient donc en y incluant ces nouveaux paramètres:

$$T(x,t) = A\exp(-\nu x)\sin(\omega t - \mu x) + B$$

On obtientdonc, pour λ, la relation suivante:

$$\lambda = \omega \rho Cp / 2\mu\nu$$

Dans cette expression les paramètres ρ et Cp sont des constantes donc le calcul expérimental de la fréquence de la sinusoïde, de μ et ν, permettra de déterminer la conductivité thermique des différents matériaux.

Les différents paramètres permettant de calculer la conductivité thermique se calculent de la manière suivante. La pulsation ω est déterminée en calculant la période de la sinusoïde. Le paramètre ν est donné en relevant l' amplitude maximale crête à crête ΔTi(xi) grâce à la relation:

$$\Delta Ti(xi) = 2\mathrm{A}\exp(-\nu\, xi)$$

Ensuite en calculant la vitesse de phase de l'onde thermique, on détermine le paramètre μ donné par la relation suivante:

$$V\phi = \omega/\mu = \delta x/\delta t = c/n$$

Pour calculer la vitesse de phase de l'onde, on se sert de la formule ci-dessus en supposant les indices de réfraction des matériaux constants malgré les variations de température. En effet, l'indice d'un milieu dépend des paramètres qui caractérisent ce milieu: température, pression, densité, ... Dans la formule, *c* est la vitesse de la lumière et vaut 3.10^8 m/s.

On prend pour l'indice du milieu les valeurs suivantes:

Acier inoxydable	Cuivre	Aluminium
n = 2,5	n = 1,10	n = 1,44

On voit donc que la seule exploitation graphique du signal sinusoïdal permet par cette méthode de calculer la conductivité thermique des matériaux acier, aluminium et cuivre. De plus on peut ainsi remarquer qu'il n'est pas nécessaire de déterminer les pertes afin d'obtenir la valeur réelle de la conductivité thermique. Ceci est dû au fait qu'on a considéré l'onde thermique comme sinusoïdale donc on remarque bien ici tout l'intérêt d'effectuer cette approximation. Dans un modèle sans pertes μ et ν sont égaux.

III. Explication sur le logiciel LabVIEW

A. Historique de LabVIEW

LabVIEW est un logiciel de développement d'applications de la société américaine National Instruments basé sur un langage de programmation graphique, le langage G. LabVIEW a été créé par Jeff Kodosky et présenté pour la première fois sur Macintosh en 1986. Le logiciel a ensuite étendu son usage au PC et est utilisable sous plusieurs systèmes d'exploitation (Windows, UNIX, Linux, Mac OS, ...). Il s'est également développé pour des applications de systèmes embarqués et temps réel. LabVIEW s'est ainsi ouvert par exemple à la programmation de circuits intégrés. La version la plus récente de LabVIEW est la 8.5.

B - Un logiciel dédié à la programmation instrumentale

Les domaines d'application traditionnels de LabVIEW sont la commande et la mesure à partir d'un PC. Ce logiciel permet donc la commande d'instruments, l'acquisition de données. Ainsi des dispositifs expérimentaux ou bancs de test peuvent être montés à l'aide de LabVIEW. Le concept d'instrument virtuel, qui a donné son nom à LabVIEW (**Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench), se manifeste par la permanence d'une interface graphique pour chaque module (fonction) d'un programme. Les contrôles et les indicateurs de ce face avant constituent l'interface par laquelle le programme interagit avec l'utilisateur (lecture de commandes et de paramètres, affichage des résultats). Les fonctions de contrôle-commande de cartes ou d'instruments constituent l'interface par laquelle le programme interagit avec le montage. Un programme LabVIEW permet donc d'automatiser un montage associant plusieurs appareils programmables, et réunit l'accès aux fonctionnalités de ce montage dans une interface utilisateur unique, véritable face avant d'un instrument virtuel.

Le fonctionnement est assuré par des bibliothèques de fonctions et des outils de développement spécialement conçus pour les applications de contrôles d'instruments et d'acquisition de données.

LabVIEW est donc un logiciel de développement d'applications comparable à la plupart des systèmes de développement en langage C ou BASIC disponibles sur le marché. Cependant ce logiciel se démarque des autres par son mode de programmation. En effet, la majorité des autres logiciels s'articulent autour de langages à base de texte dont la programmation consiste à empiler des lignes de code, tandis que LabVIEW utilise un langage de programmation graphique, le langage G, pour créer un programme sous forme de diagramme.

C. Avantages de LabVIEW

Ce mode de programmation est le principal avantage de LabVIEW. Ainsi, il est inutile d'être expert en programmation pour utiliser ce logiciel. La terminologie, les icônes et les principes inhérents à LabVIEW, tous familiers aux ingénieurs et aux scientifiques, font appel à des symboles graphiques pour décrire les opérations de programmation. Ainsi le langage graphique par flux de données de LabVIEW et l'approche par diagramme, qui représentent naturellement le flux de données et conçoivent intuitivement l'interface utilisateur avec tous les contrôles de données, permettent de voir facilement et de modifier les données ou les entrées de contrôle. Pourtant, bien que plus intuitif, LabVIEW fournit les performances, la souplesse et la compatibilité des langages de programmation classiques (C, BASIC, ...). En effet, le langage G contient les mêmes éléments que les langages classiques: des variables, des types de données, des boucles, des structures de séquence et la gestion d'erreurs.

Un second avantage est offert par des bibliothèques étendues de fonctions et de routines (blocs pré-programmés) capables de répondre à la plupart des besoins en programmation. LabVIEW contient également, sous plusieurs plates-formes dont Windows, des bibliothèques de fonctions spécifiques à l'acquisition de données et au pilotage d'instruments VXI et GPIB, ou encore d'instruments connectés sur une simple liaison série. Il existe aussi des bibliothèques dédiées à la présentation, à l'analyse et au stockage des données. LabVIEW intègre une panoplie complète d'outils de développement de programme conventionnels, ainsi des points d'arrêt, des sondes et des animations d'exécution, qui mettent en évidence le cheminement des données et des exécutions pas à pas du programme, peuvent être utilisés afin de déboguer les programmes.

D. Comment marche LaVIEW?

Un programme LabVIEW est appelé Instrument Virtuel ou VI tout simplement parce que sa représentation et son fonctionnement ressemblent à ceux d'instruments classiques. Néanmoins, les VIs diffèrent en ce sens qu'ils tirent leur fonctionnalité de la programmation informatique.

Ces VIs comportent trois composants: la face avant, la fenêtre diagramme et l'icône et ses connecteurs:

- la face avant est une interface utilisateur intéractive. Elle simule la face avant d'instruments physiques et contient des commandes (boutons-poussoirs, ...) et des indicateurs (jauges, graphe, ...). Des données peuvent ainsi être saisies afin de visualiser les résultats à l'écran.
- un VI reçoit des instructions de son diagramme, qui est construit en langage graphique. Le diagramme, qui correspond au code source du VI, réduit ainsi la programmation à une simple manipulation graphique.
- le cadre icône/connecteur d'un VI répertorie sous forme graphique tous ses paramètres.

Un VI présente une structure hiérarchique et modulaire. Il peut être utilisé comme un programme principal ou comme un sous-programme à l'intérieur d'autres programmes ou de sousprogrammes. Un VI contenu à l'intérieur d'un autre VI s'appelle un sous-VI. Il permet l'utilisation d'un code récurrent dans le programme principal et comme il utilise moins de mémoire, alors un sous VI permet de déboguer plus facilement un programme.

De plus, LabVIEW possède la particularité d'être multitâche, ainsi plusieurs parties de code peuvent s'exécuter simultanément.

IV. Algorithme complet du programme

Avant de commencer à programmer sous LabVIEW, des algorithmes de fonctionnement de l'application ont été conçus afin de simplifier la programmation et d'avoir une meilleure vision du problème.

A. Algorithme général

Le programme nécessite d'être multitâche puisque l'acquisition T=f(t) doit pouvoir s'exécuter en même temps que l'on traite les données pour les différents calculs et en même temps que l'on chauffe la barre. Pour ce faire, on utilise 4 tâches qui sont:

- une gestion interface
- une acquisition Ti=f(t)
- une acquisition Ti=f(x) avec le calcul de λ
- un échauffement de la barre

En effet, une seule tâche serait trop contraignante pour l'utilisation du programme.

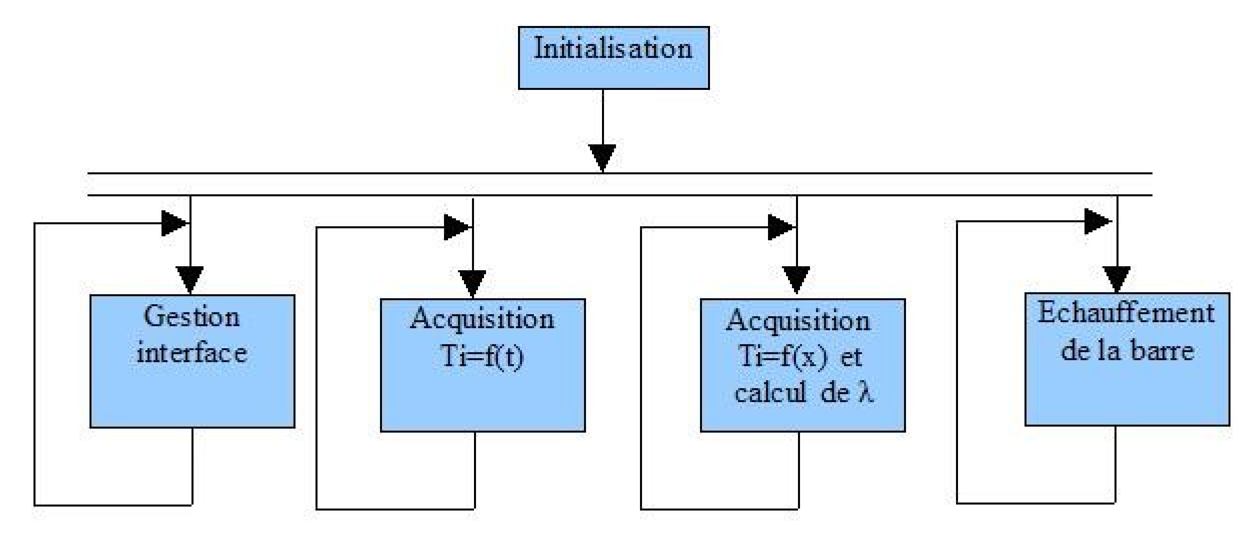


Illustration 1: algorithme général de fonctionnement

B. Sous algorithmes

Dans cette partie, les algorithmes des quatre parties principales du programme seront dévoilés sous forme de grafcet. L'algorithme de la partie "échauffement de la barre" est le suivant:

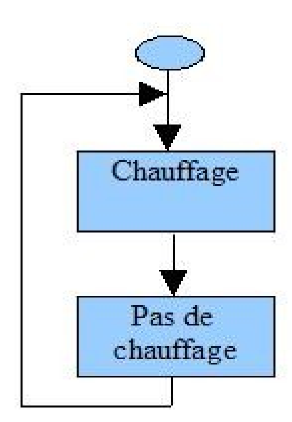


Illustration 2: algorithme de la fonction échauffement de la barre

Pour la fonction "acquisition Ti=f(t)", l'algorithme est donné sur la page suivante pour des soucis de lisibilité. Cette partie du programme consiste à acquérir les huit différentes courbes temporelles en température lors du chauffement des barres. Ces températures sont fournies par des thermocouples disposés tous les 10cm afin d'étudier le gradient température le long des différentes barres. Les valeurs de températures sont ensuite affichées dans un tableau excel afin de pouvoir exploiter les résultats et de tracer les courbes Ti=f(x).

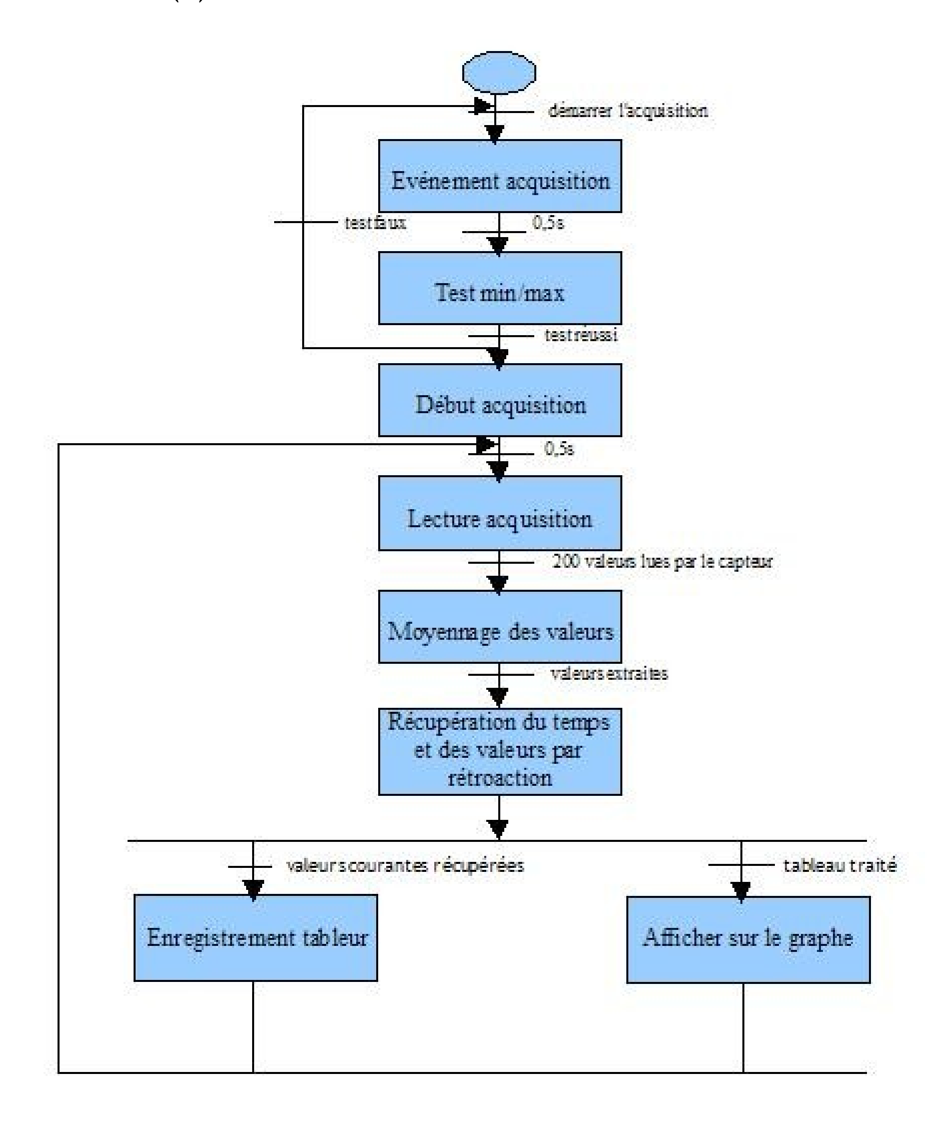


Illustration 3: algorithme de "acquisition Ti=f(t)"

L'algorithme de la fonction permettant l' "acquisition de Ti=f(x) et le calcul de λ" est dévoilé à la page suivante:

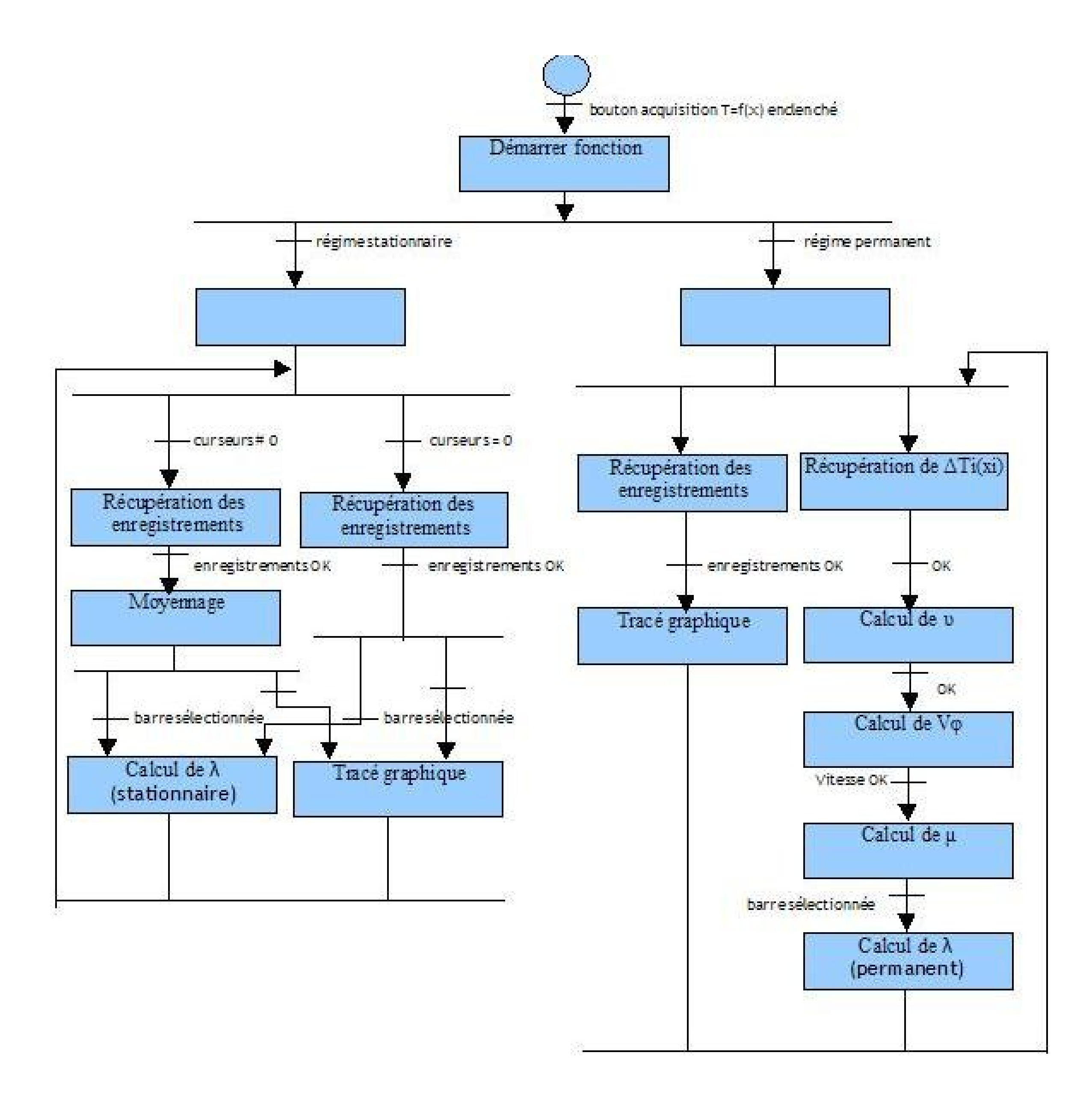


Illustration 4: algorithme de la fonction "acquisition de Ti=f(X) et calcul de λ"

En ce qui concerne la gestion de l'interface, l'algorithme est simple, il effectue l'action de mise à jour de celle ci dès que l'on appuie sur un bouton ou que l'on effectue une action quelconque. Ces actions sont :

- L'arrêt de Ti=f(t)
- L'arrêt de T=f(X)

- L'arrêt de la chauffe
- Le reset graphique 1
- Le reset graphique 2
- La sélection du régime
- Le choix d'un menu
- La sortie de l'application

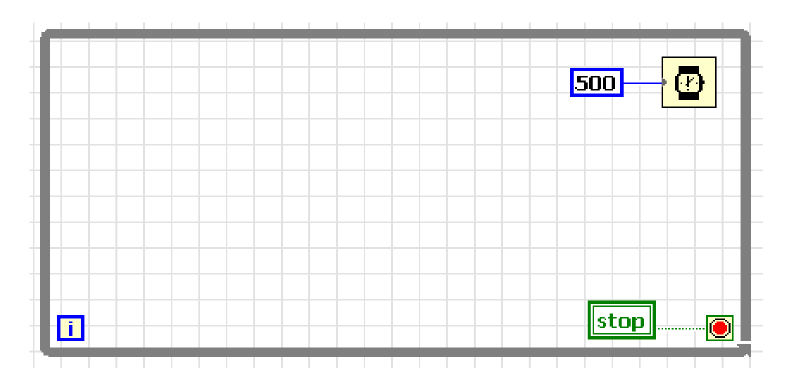
Il est important de rajouter que la gestion d'erreurs ainsi que la gestion d'arrêt des tâches ne sont pas représentés sur les algorithmes dans un souci de lisibilité.

V. Description des parties importantes du programme

A. Principes de base

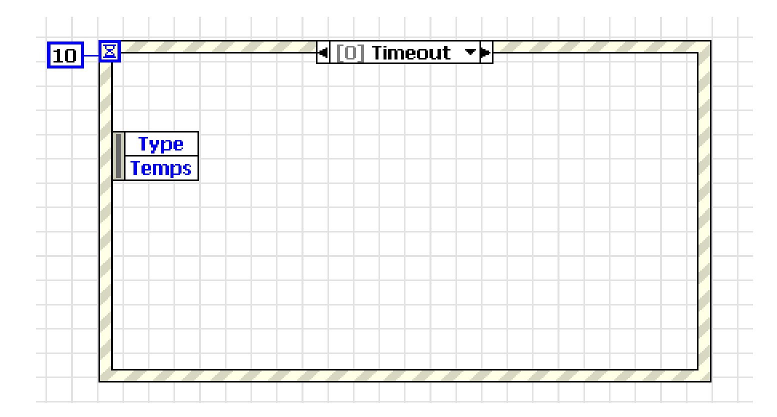
LabVIEW est un langage de programmation qui a la particularité d'être facilement manipulable visuellement parlant, ainsi le développement s'en voit simplifié. Nous allons donc décrire les différentes structures les plus utilisées pour le développement de notre programme :

- La structure "boucle while" permet d'effectuer une action indéfiniment suivant une condition :



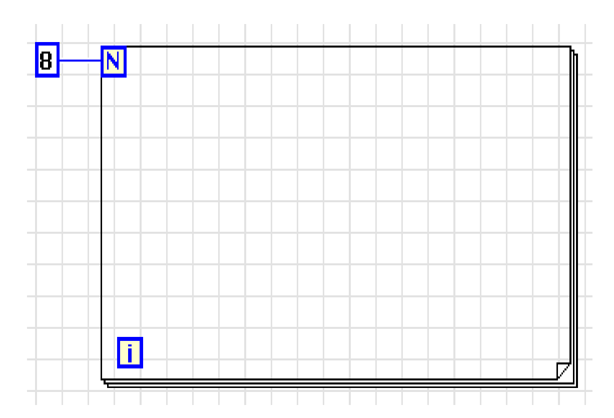
Ici, la boucle while s'exécute successivement toutes les 500ms et s'arrête sur la condition stop. Dans notre programme, on associe chaque boucle while à une tâche précise ou fonction qui se répète.

- La structure "événements" permet de gérer des actions qui se déroulent sur l'interface utilisateur telle que le clic de souris :



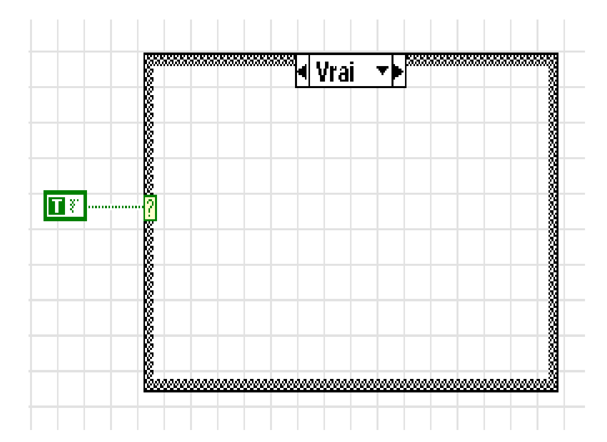
Ici, la structure gère l'événement timeout et exécute le code qui s'y trouve. Dans notre programme, on associe surtout les événements à des clics sur les boutons.

- La structure "boucle for" permet d'effectuer n fois les mêmes actions avec n donné :



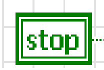
Ici, la boucle for s'exécutera 8 fois (l'indice i donne l'indice de la boucle de 0 à 7 ici). Dans notre programme, les structures for sont surtout utilisées pour couvrir les traitement de données des 8 capteurs.

- La structure "condition" est de loin la structure la plus utilisée car il est souvent nécessaire qu'une action soit contrainte par une condition, elle prend juste une variable en entrée tel qu'un booléen ou encore un entier et permet d'effectuer différentes actions suivant la valeur de l'entrée :



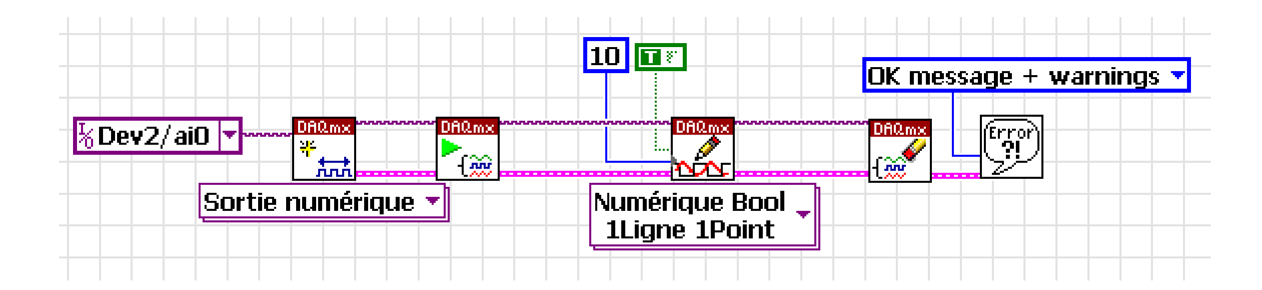
Ici, le paramètre d'entrée est "vrai" donc l'action dans la partie vrai de la condition sera exécuté. Dans notre programme, il y a beaucoup de conditions pour les calculs ou encore pour les contrôles.

- La variable locale est utile si l'on veut se servir d'un contrôle plusieurs fois :



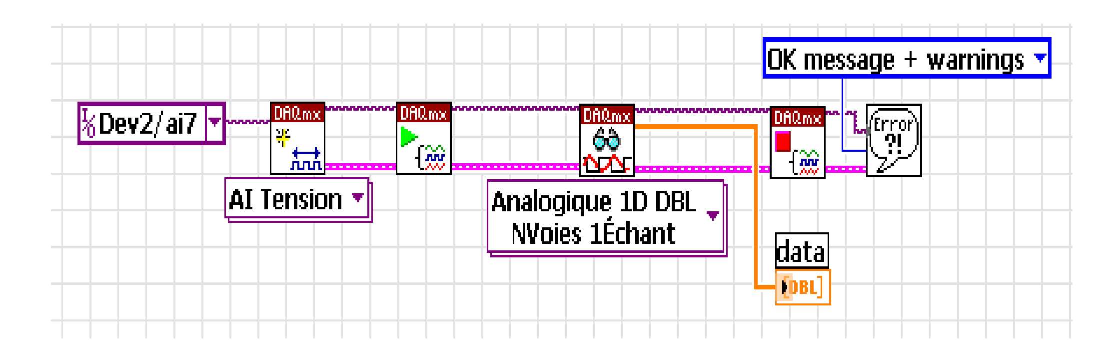
On peut voir ici la représentation de la variable locale du contrôle stop.

- Dialogue avec l'appareil de mesure/chauffe :
  - Ecriture d'une valeur :



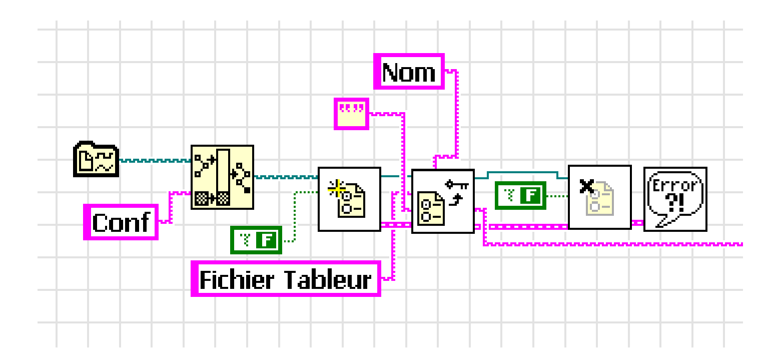
Pour écrire une valeur, on sélectionne le canal, crée une voie, démarre une tâche, écrit une valeur, réinitialise la tâche et on gère l'erreur. On remarque ici que l'on écrit une valeur numérique en sortie.

- Lecture de valeurs :



Pour lire, c'est le même principe sauf que l'on remplace la fonction lire par écrire (on peut lire n voies car on a n capteurs). Par contre, on lit ici une valeur analogique en entrée (tension).

- Lecture/écriture d'une clé de configuration :

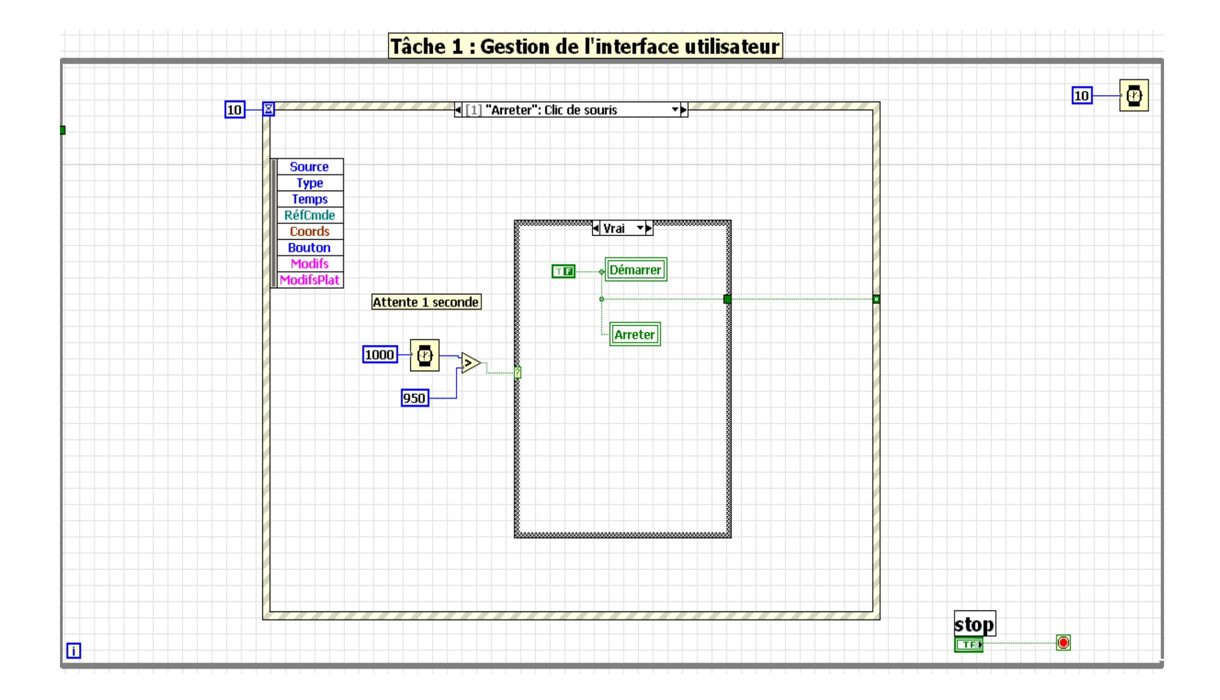


On récupère le chemin par défaut, on recompose avec le nom du fichier de configuration, on ouvre les données de configuration, on lit la clé de la chaine fichier tableur à la section Nom, on ferme ce fichier et on gère l'erreur. Puis pour l'écriture, il suffit juste de remplacer le bloc lecture clé.

B. Détails des différentes tâches

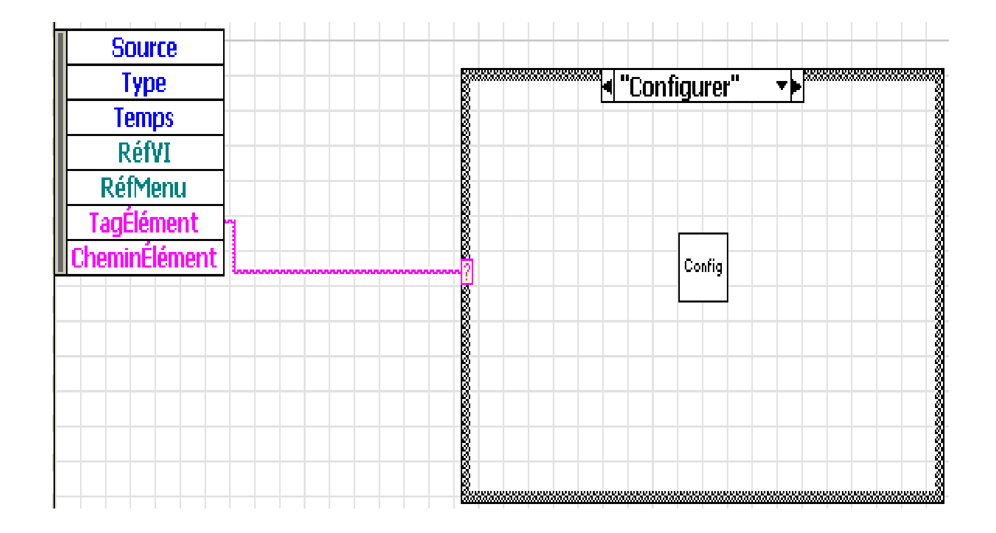
- Tâche de gestion de l'interface utilisateur

Cette tâche est gérée par une structure évenements imbriquée dans une boucle while.



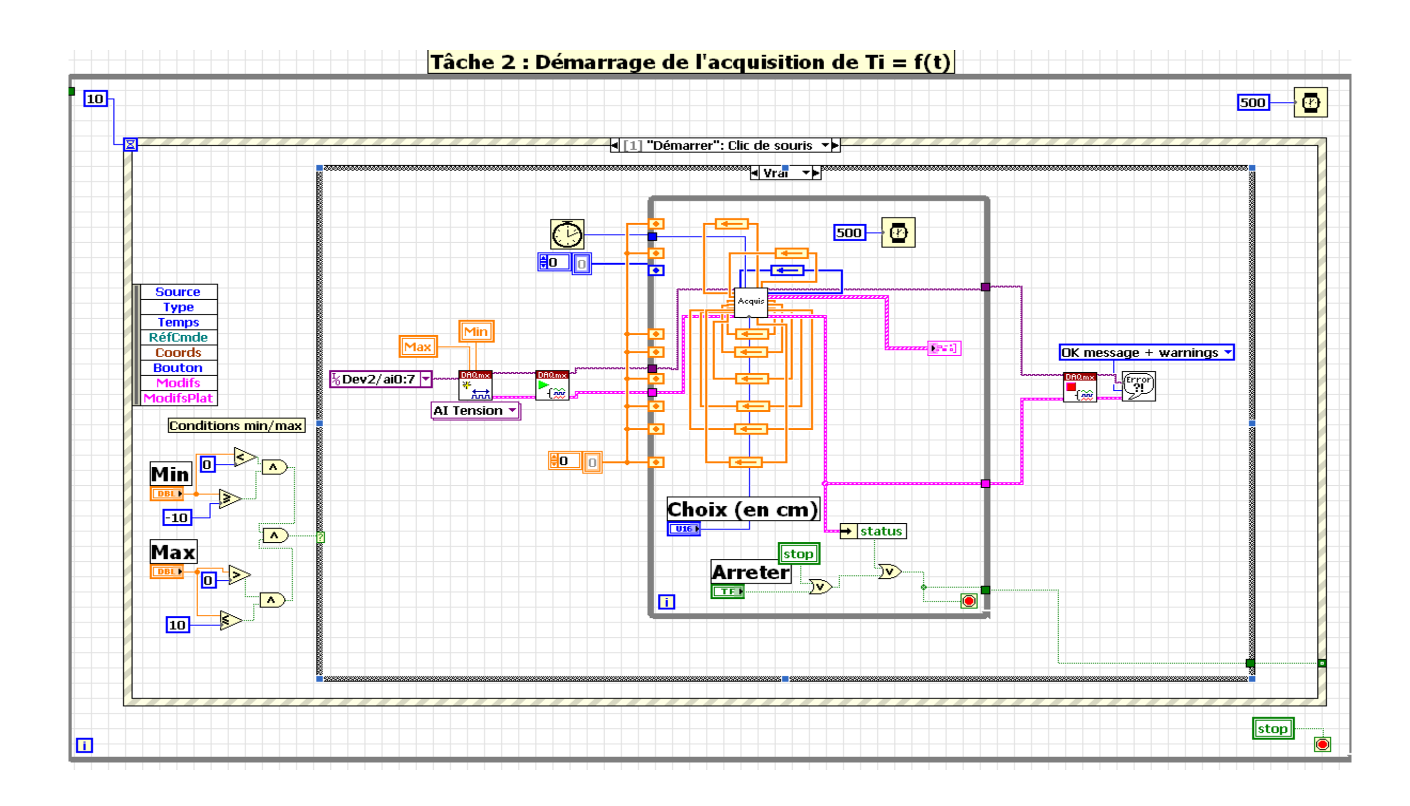
Lorsque l'on clique sur arrêter, on attend un peu moins d'une seconde et on remet les contrôles Démarrer et Arrêter à Faux pour pouvoir les réutiliser. On gère ainsi Arrêter acquisition Ti = f(t), Arrêter T = f(X) + calculs, Fin chauffe, Reset graphique 1 et Reset graphique 2.

Dans le cas de la sélection du régime, c'est différent puisque l'on utilise l'option sortie de souris pour mettre à jour la face avant. L'évenement "Face avant fermée ?" est aussi utilisé pour quitter l'application. De plus, le cas du menu est intéressant car il permet de rendre l'interface plus conviviale :



On lance ici un sous VI config si le tag d'élément vaut "Configurer".

- Tâche d'acquisition Ti = f(t)

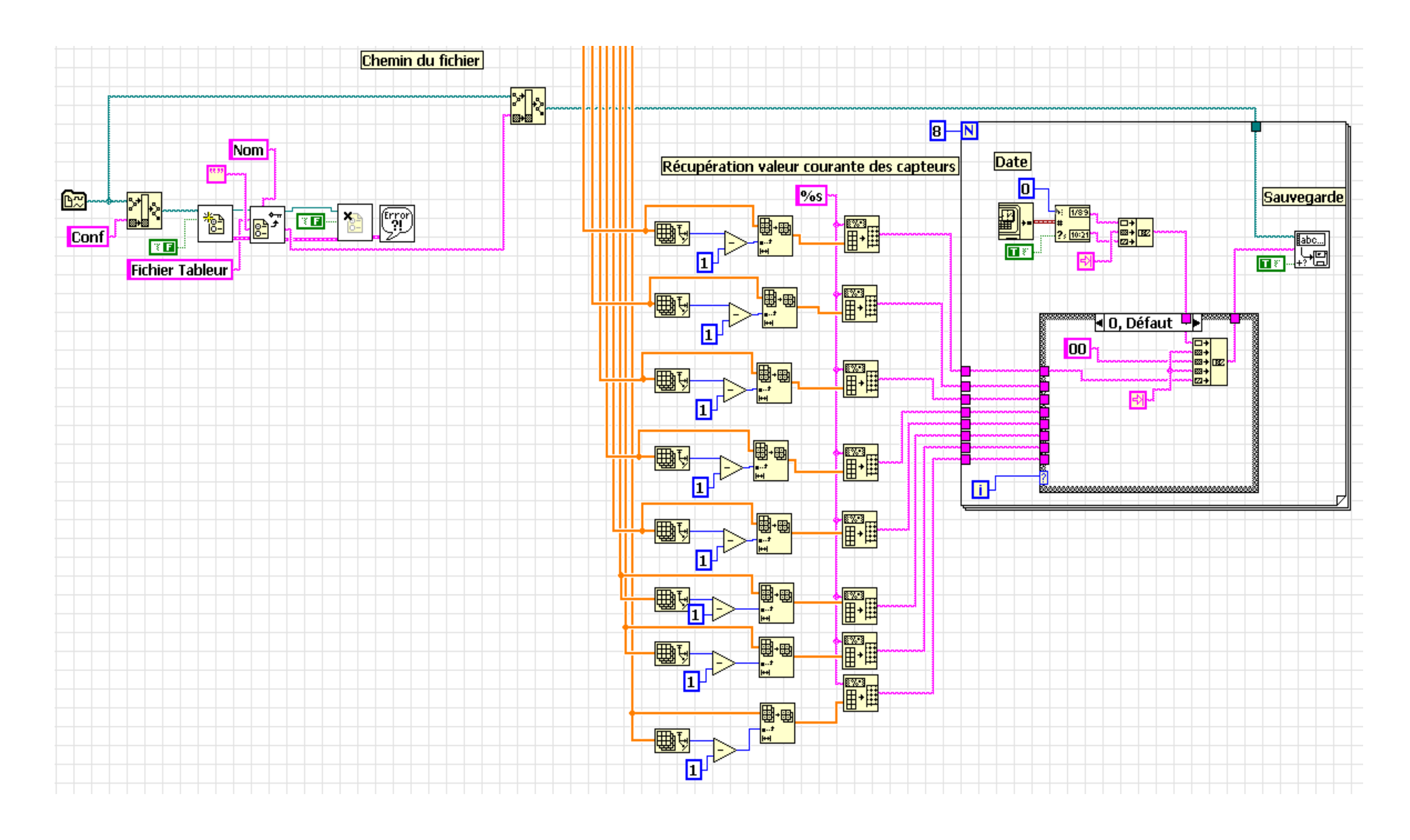


On remarque la présence d'une structure événement affiliée au clic sur Démarrer dans une boucle while. Il y a tout d'abord le test avec une condition puis la sous partie dite d'acquisition "time software" puisque l'on utilise le temps du logiciel et de l'ordinateur. La sous partie d'acquisition fonctionne sur le principe de la lecture de valeurs sur l'appareil précédemment vu.

Pour l'acquisition time software, on concatène le tableau de valeurs temporelles précédentes avec la valeur courante du temps. De la même façon, on concatène le tableau de chaque mesure capteur précédente avec la valeur courante de chaque capteur.

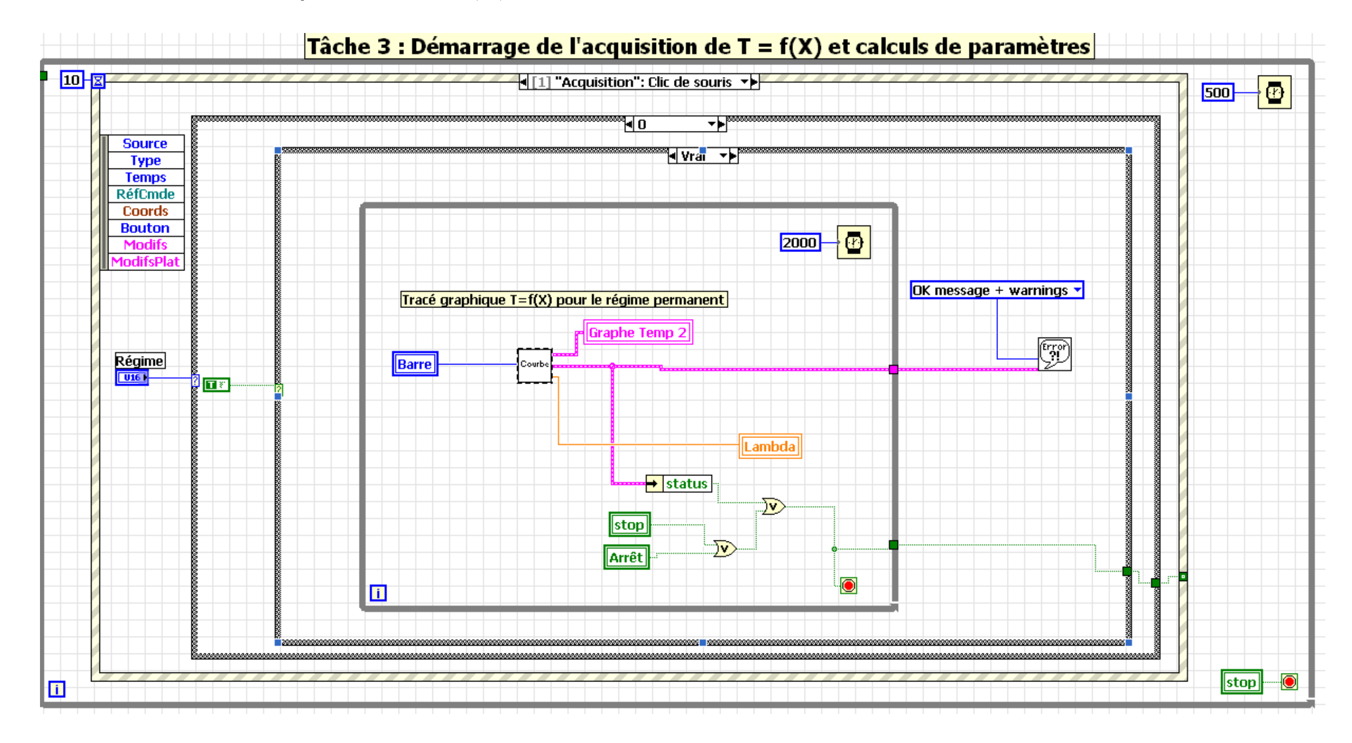
Sinon, la mesure s'effectue pour chaque capteur avec un moyennage sur 200 échantillons prélevés afin de supprimer une partie des bruits résiduels mais en pratique il y a tout de même beaucoup de bruit par rapport aux capteurs de températures. On vient ensuite lire la valeur du tableau récupérée avec le bon indice et on rentre cette valeur ainsi que le temps dans un cluster afin de pouvoir l'exploiter pour l'affichage du graphique.

Il est utile d'enregistrer les valeurs dans un tableur :

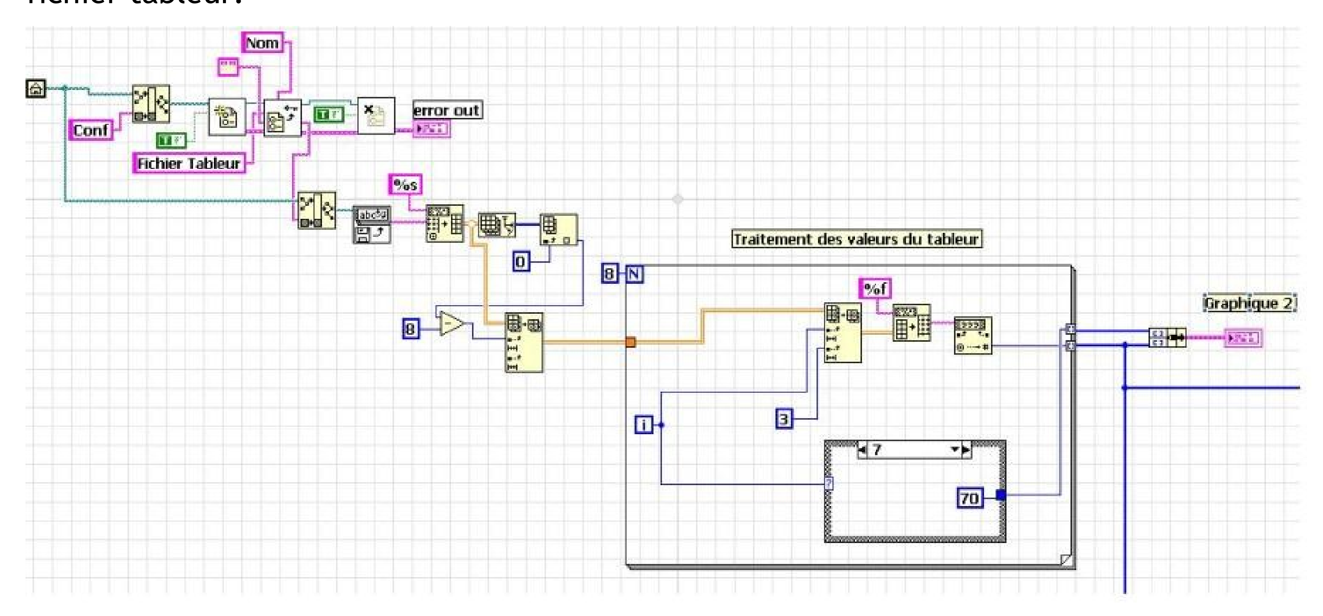


Ici, on extrait les valeurs capteurs des tableaux et on effectue 8 enregistrements successifs pour les 8 capteurs. Un enregistrement est composé de la date, de la distance du capteur et de sa valeur. Le fichier tableur a pour extension .txt mais on peut l'ouvrir avec microsoft excel.

- Tâche d'acquisition T=f(X) et différents calculs

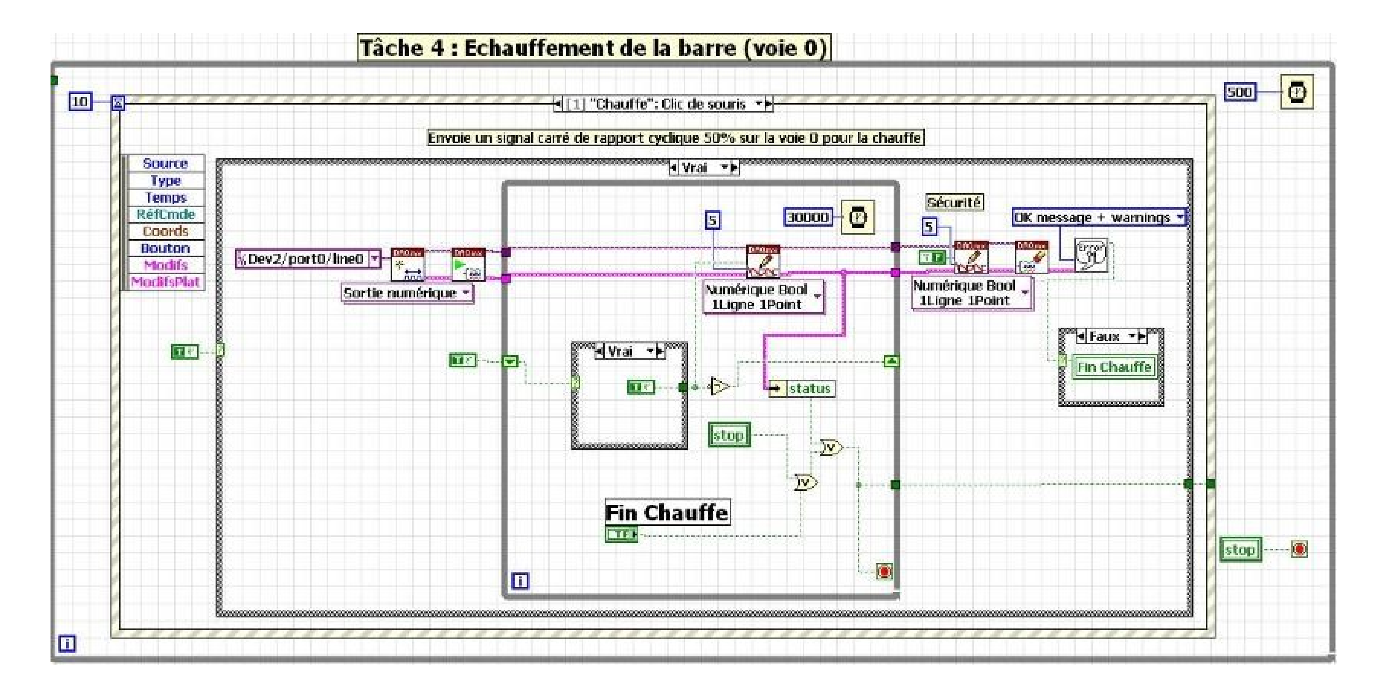


Pour cette tâche, on teste la sélection régime (stationnaire ou permanent). En ce qui concerne les curseurs, on fait un moyennage sur toutes les valeurs comprises dans l'intervalle des deux curseurs afin de tracer T=f(x). Prenons le cas du régime stationnaire; pour tracer la courbe T=f(x), il nous faut extraire les valeurs capteurs du fichier tableur:



On lit après les valeurs des 8 capteurs du dernier enregistrement du tableur et on trace la courbe de T=f(X). On calcule ensuite λ qui vaut -j/pente. Le calcul préalable de la pente est soumis à une méthode qui conserve la partie stable de celle ci.

- Tâche pour la chauffe à 50%



On utilise ici le principe de l'écriture des valeurs sur l'appareil mais 1 fois toutes les 30 secondes. On utilise un registre à décalage afin de pouvoir modifier l'état de la donnée écrite et ainsi, on envoie un signal carré de rapport cyclique 50% sur la voie 0 de l'appareil.

En résumé, le programme se compose de:

- Main.vi pour l'application principale
- Acquisition.vi pour Ti=f(t)
- Courbe.vi pour T=f(X) en stationnaire
- Courbe2.vi pour T=f(X) en permanent
- Config.vi pour configurer
- Conf qui est le fichier d'enregistrement de la configuration
- Enregistrement.txt qui est le tableur

VI. Manuel d'utilisation

A. Introduction et vue d'ensemble

Ce programme est un programme dédié à l'étude de la propagation d'une onde thermique dans une barre. Il possède différentes fonctionnalités que nous allons étudier mais d'abord nous allons présenter l'interface :

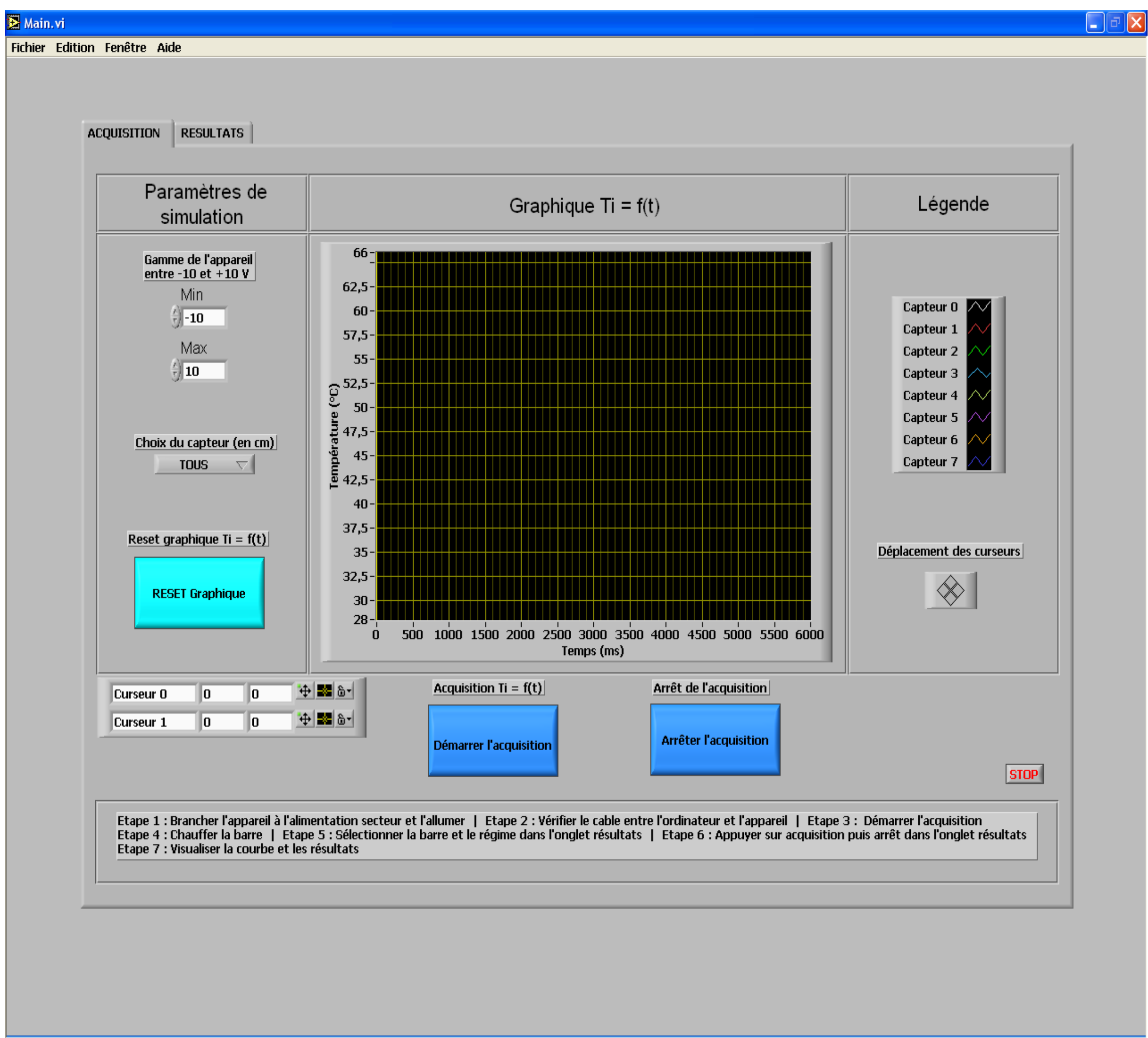


Illustration 5: interface principale

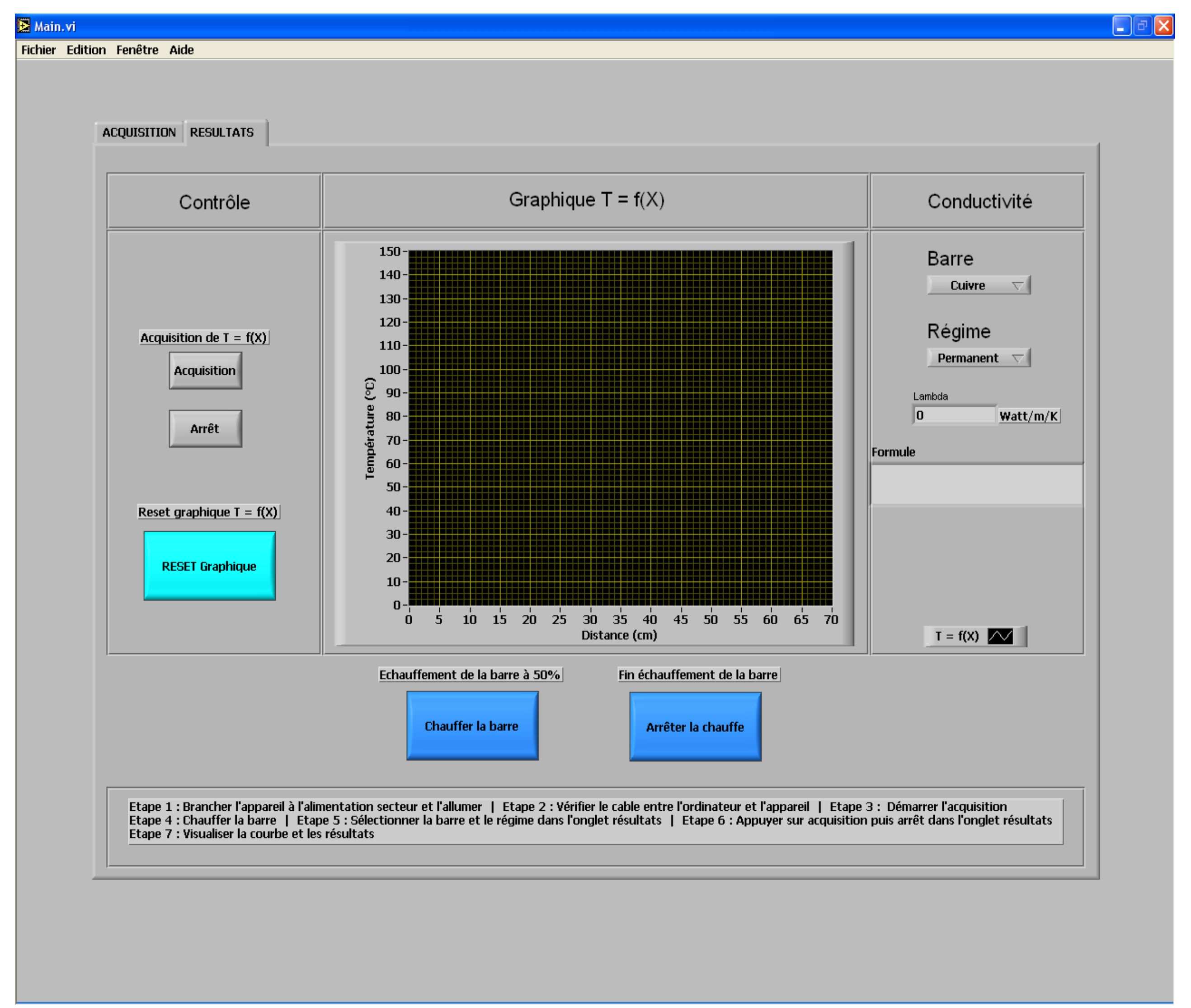


Illustration 6: interface secondaire

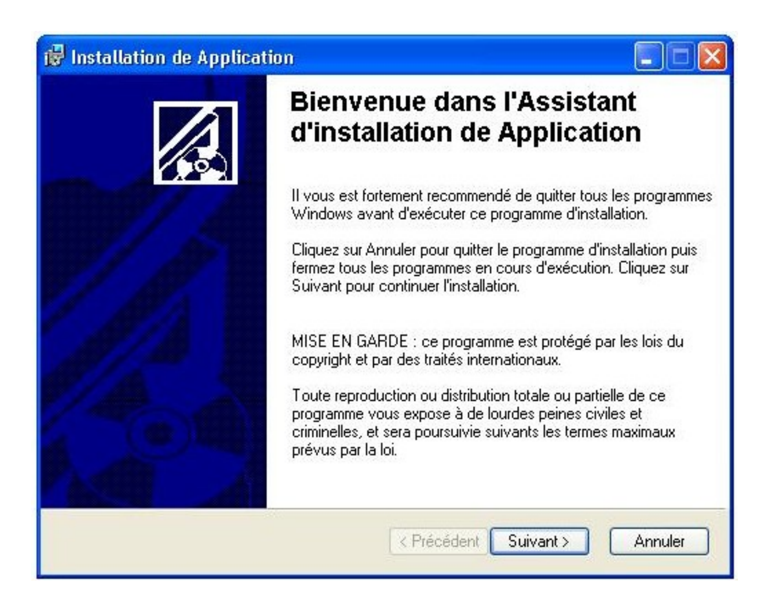
B. Démarrage

Installation

Procédez comme suit pour installer le programme sur votre ordinateur :

1. Allumez votre ordinateur.

2. Lancez l'exécutable situé dans le dossier d'installation fourni. Le menu devrait s’afficher.



3. Cliquez sur suivant et laissez-vous guider par l'assistant d'installation du logiciel. Une fois que vous avez créé un dossier pour le programme, le programme est installé sur votre disque dur.

4. Après avoir installé le programme, sélectionnez "Programme etudiant / Acquisition température " à partir du menu Démarrer de votre ordinateur ou cliquez à deux reprises sur le raccourci de Acquisition température s'il est visible sur votre bureau. Les deux opérations ont strictement le même résultat.

Désinstallation

Procédez comme suit afin de désinstaller le programme :

1. Allumez votre ordinateur.

2. Sélectionnez le programme d'Ajout/suppression de programmes de votre panneau de configuration et désinstaller Acquisition température.

3. L'assistant de désinstallation vous demande alors si vous souhaitez désinstaller le programme. Si vous êtes sûr de vouloir le faire, cliquez sur Oui. Dans le cas contraire, cliquez sur Non, ce qui interrompt le processus de désinstallation.

C. Interface principale

- Démarrer l'acquisition: Démarrage de l'acquisition Ti=f(t) (T en °C et t en ms).
- Arrêter l'acquisition: Arrête l'acquisition Ti=f(t).
- Reset graphique : Remise à zéro du graphique (le graphique est bien sûr reinitialisé à chaque lancement d'acquisition).
- Déplacement des curseurs : Pas encore implémenté dans le programme.
- Choix du capteur : Affichage dynamique de la courbe en température du capteur sélectionné. Attention : les enregistrements sont toujours valables pour tous les capteurs.

D. Interface secondaire

- Acquisition : Démarrage de l'acquisition T=f(X) (T en °C et X en cm).
- Arrêt : Arrête l'acquisition T=f(X).
- Reset graphique : Remise à zéro du graphique (le graphique est bien sûr réinitialisé à chaque lancement d'acquisition).
- Chauffer barre : Chauffe la barre alternativement.
- Arrêter la chauffe : Stop la chauffe de la barre.
- Barre : Ajuste les calculs pour la barre.
- Régime : L'utilisateur choisit le mode d'acquisition.

E. Menu Configuration

- Nom du fichier : Il faut rentrer le nom du fichier (exemple : fichier.txt) seulement si l'on souhaite changer de fichier sinon ce n'est pas la peine.
- Créer fichier : Si l'on veut créer un fichier du nom choisi.

F. Utiliser le logiciel

Etape 1 : Brancher l'appareil à l'alimentation secteur et l'allumer

Etape 2 : Vérifier le cable entre l'ordinateur et l'appareil

Etape 3 : Démarrer l'acquisition

Etape 4 : Chauffer la barre

Etape 5 : Sélectionner la barre et le régime dans l'onglet résultats

Etape 6 : Appuyer sur acquisition puis arrêt dans l'onglet résultats

Etape 7 : Visualiser la courbe et les résultats

G. Assistance technique

• Vous éprouvez des difficultés lors des phases d'installation, de lancement ou d'exécution de votre logiciel ?

• Vous désirez être informé des patches et mises à jour concernant le produit ?

N'hésitez pas à contacter notre Support Technique qui vous donnera la bonne marche à suivre pour que vous puissiez exploiter votre logiciel comme il se doit.

mail : gendre@polytech.unice.fr, leca@polytech.unice.fr

Conclusion

L'utilisation de LabVIEW, par son langage de programmation graphique et modulaire, a simplifié le code du programme. De plus ce mode de programmation est bien plus accessible que les langages textuels classiques comme le C/C++ ou le JAVA par exemple. Ainsi, la programmation de ce banc de test est bien plus intuitive en utilisant LabVIEW.

De plus, la présence de drivers DAQ permet de communiquer facilement, par le biais des entrées/sorties analogiques et numériques, avec l'environnement du TP. L'acquisition de données, qui est primordiale dans ce projet, a été faite par une acquisition de type "time software" de la lenteur des mesures (type DC). Cependant, le résultat concernant la conductivité thermique en régime permanent est abérrant, cela

doit être du au calcul de la vitesse de phase.

Enfin, du fait de l'imprécision des thermocoupleurs et des perturbations extérieures, les mesures peuvent être légèrement faussées. Ainsi, le résultat obtenu pour la conductivité thermique donne juste un ordre de grandeur du paramètre λ.

Ce banc de test d'un TP de thermodynamique a été conçu dans le but de simplifier la tâche d'un étudiant se trouvant face au système. Ainsi, l'interface ainsi qu'un manuel de fonctionnement expliquent à l'étudiant les différentes étapes afin de mener à terme le TP.